

National Aeronautics and Space Administration



Visualization of Monitoring Data at the NASA Advanced Supercomputing Facility

Janice Singh

janice.s.singh@nasa.gov

www.nasa.gov



NASA Advanced Supercomputing Facility

- Pleiades
 - 11,136-node SGI ICE supercluster
 - 162,496 cores total (32,768 additional GPU cores)
- Tape Storage
 - pDMF cluster
- NFS servers
 - home filesystems on computing systems
- Lustre Filesystems
 - each filesystem is multiple servers
- PBS (Portable Batch System)
 - job scheduler that runs on computing systems

Ref: <http://www.nas.nasa.gov/hecc/>

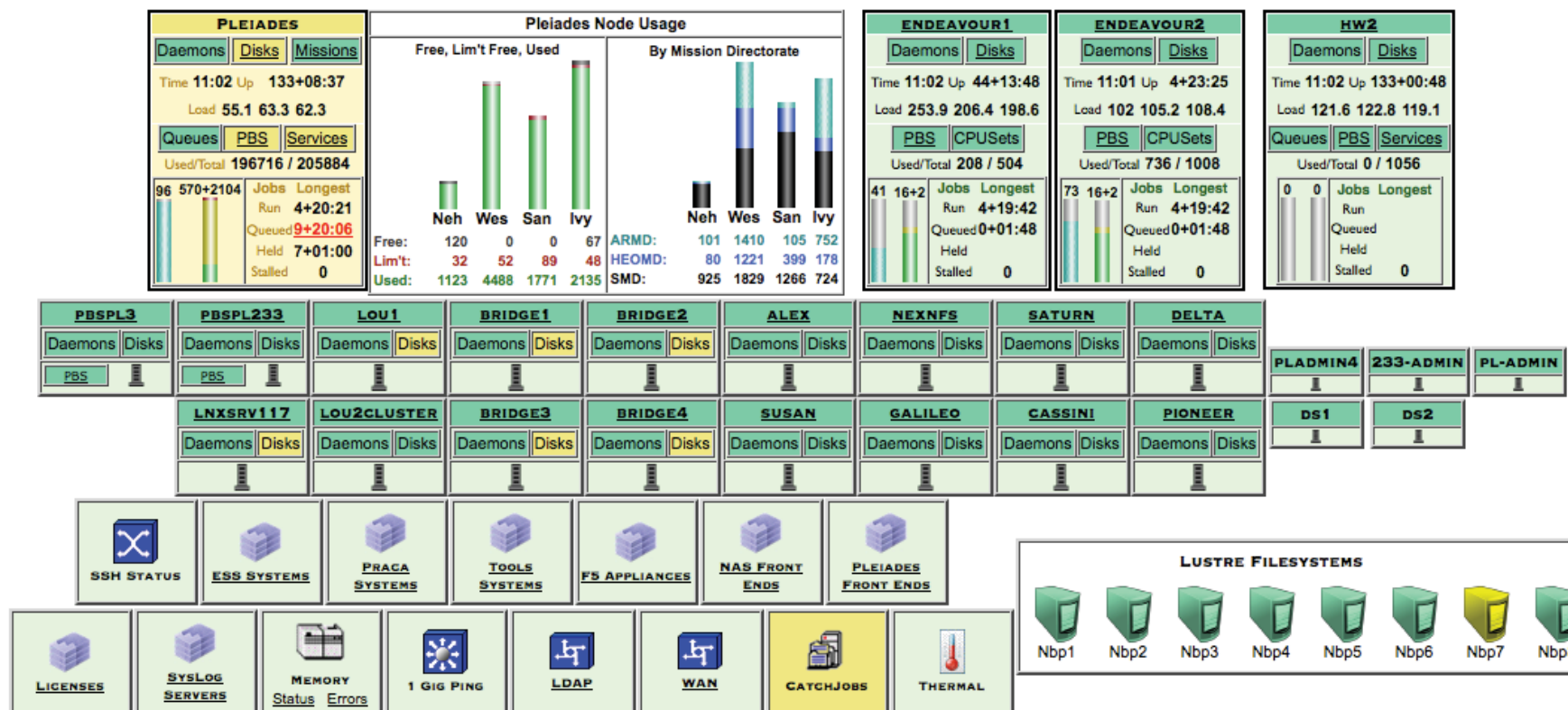


Why Visualization is Needed

- 24 x 7 Help Desk
 - need a quick overview of system status
 - but still more specific than nagios visualization
 - not just single status per host, but sub-groups per host
 - they assess situations before calling next level of support
 - automatic alerts from nagios are not as selective
 - interrelated issues
 - allows us to see how many systems affected by one issue

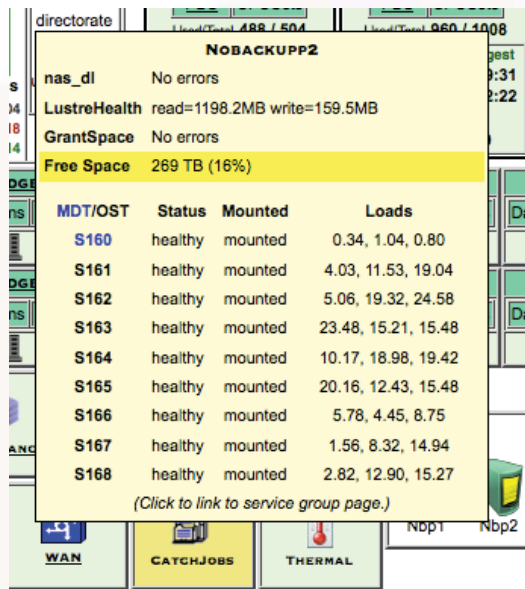
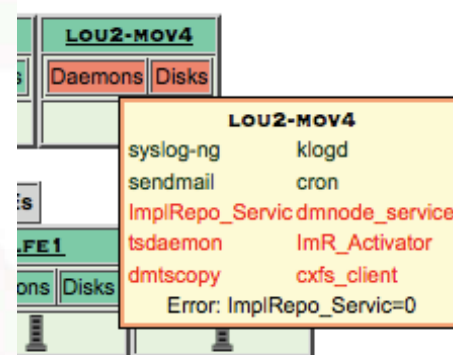
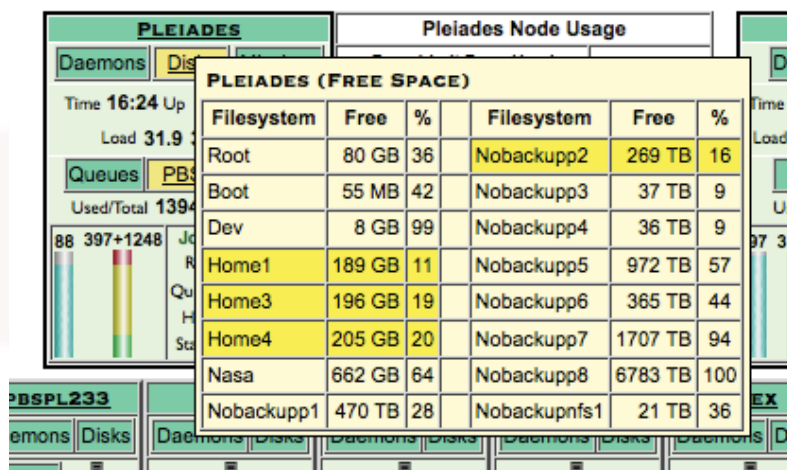


Heads Up Display – For Staff



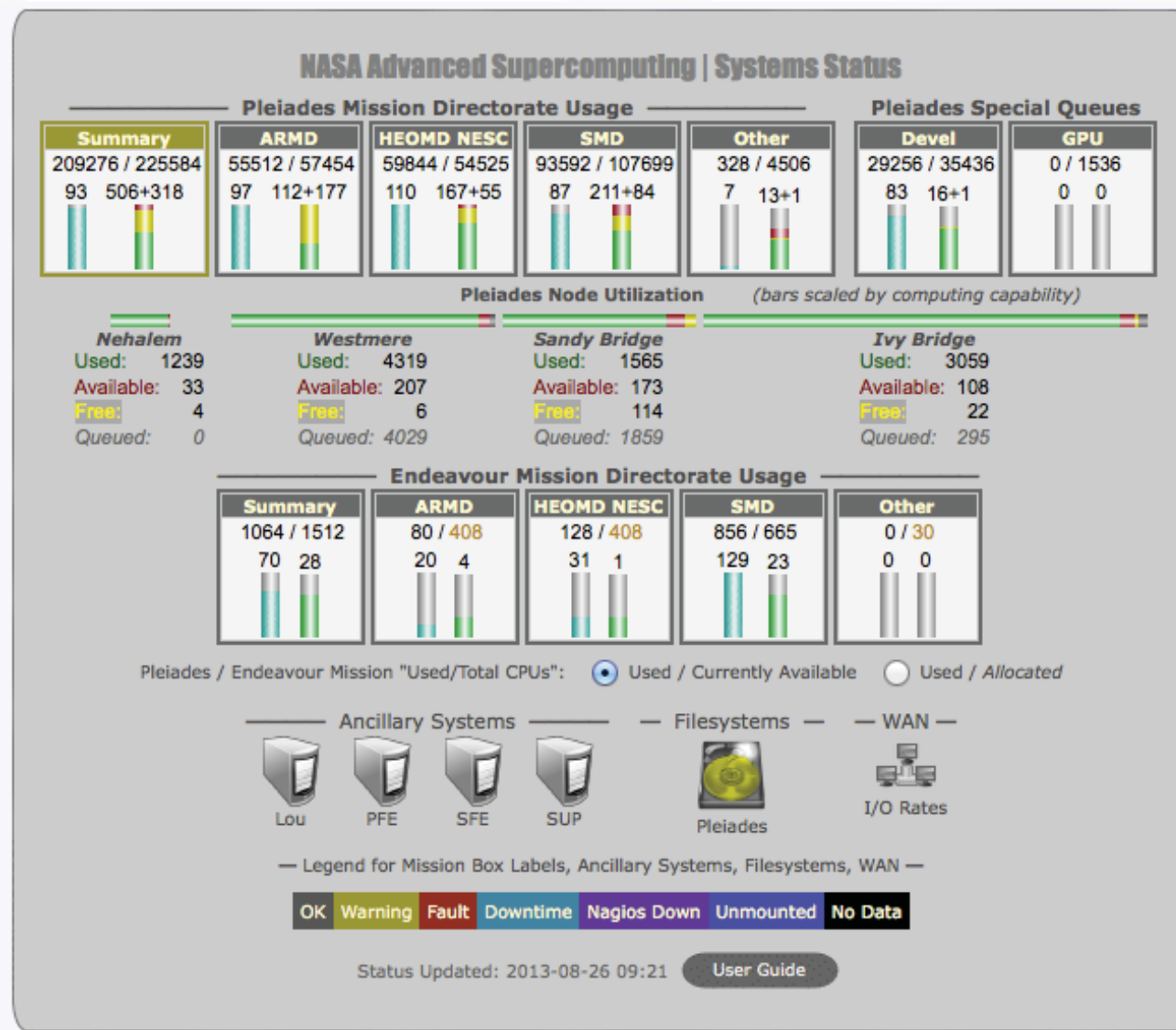


Heads Up Display – For Staff (details)





Heads Up Display – for Users



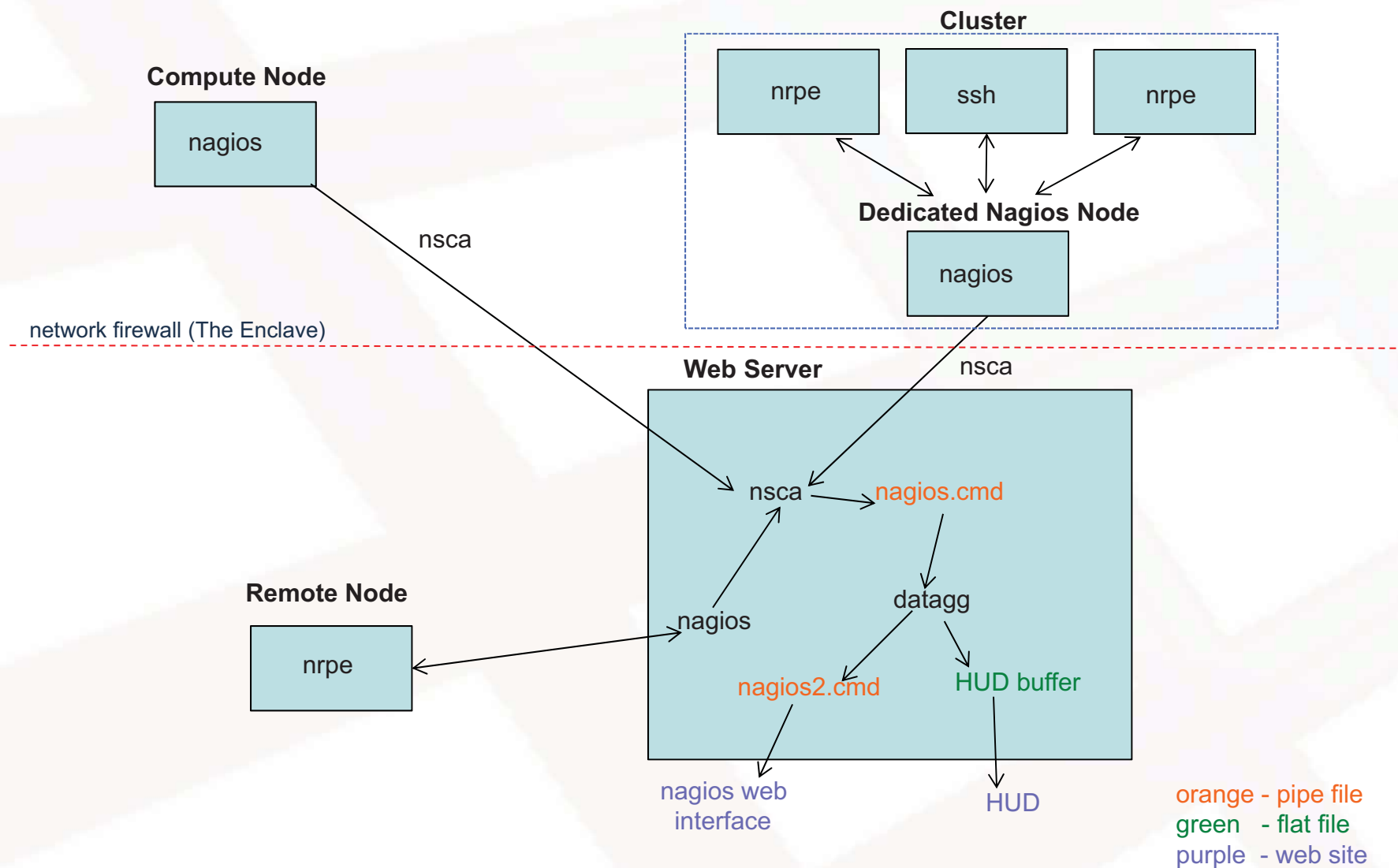


All the Parts

- nagios
- nrpe
- nsca
- datagg (in-house software)
- apache
- perl/cgi-bin



Data flow





Nagios (and add-ons) setup

The Basics

- the webserver and the main nagios server are the same machine
- there is a network firewall called “the enclave”
 - most compute nodes are inside the enclave
 - the webserver can only receive data from the enclave, not send
- the servers within the enclave send nagios data to the webserver via nsca
- for the servers outside the enclave, nrpe is used
- plugins written in Perl using Nagios::Plugin



Nagios (and add-ons) setup, cont.

Versions

- when I inherited the systems, they were all using nagios 2.10
 - most systems have been upgraded to nagios 3.4+
 - all new systems have nagios 3.5
 - webserver is still using 2.10
- nsca 2.7.2 across the board



Nagios (and add-ons) setup, cont.

Clusters within the enclave

- there is one host that is considered the Dedicated Nagios Server
- the rest of the hosts are monitored using nrpe
- exceptions on Pleiades:
 - there are many hosts monitored that get reimaged often
 - difficult to administer nrpe
 - use check_by_ssh
 - ssh is flaky under nagios 3, so it still uses nagios 2.10
 - will randomly give the error: Could not open pipe: /usr/bin/ssh
 - use 2 Dedicated Nagios Servers
 - so many checks that there was unacceptable latency
 - tests that should run every 2 mins were running every 30 mins



datagg (DATa AGGregator)

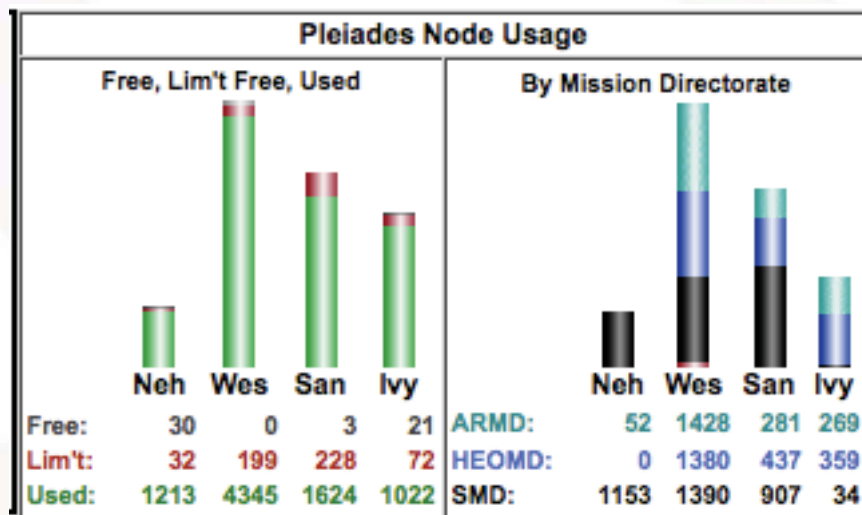
- why it was needed (i.e. what nagios couldn't do for us)
 - error summaries of nagios problems
 - the nagios webpage does tell number of alerts per service group, but that cannot be leveraged via API

PLEIADES			Pleiades Node Usage		
ns	Disks	Missions	Free	Lim't	Free Used
PLEIADES (FREE SPACE)			By Mission Dire		
1:42 Up			Filesystem	Free	%
141.2 13			Root	80 GB	36
s PBS			Nobackupp2	384 TB	23
total 17033			Boot	55 MB	42
1948 Job			Nobackupp3	32 TB	8
Run			Nobackupp4	57 TB	14
Queue			Nobackupp5	980 TB	58
Hel			Nobackupp6	359 TB	43
Stall			Nobackupp7	1709 TB	94
PBSPL23			Nasa	655 GB	64
emons D			Nobackupp8	6776 TB	100
85			Nobackupp1	454 TB	27
			Nobackupnfs1	16 TB	27



datagg (DATa AGGregator), cont.

- why it was needed (i.e. what nagios couldn't do for us)
 - parsing data about Portable Batch System (PBS)



- piecing together large outputs from NSCA
 - current output from PBS: 404,937 characters



datagg (DATa AGGregator), cont.

- why it was needed (i.e. what nagios couldn't do for us)
 - mapping service nodes to the appropriate Lustre filesystem
 - they are in a servicegroup (but not available via API)

Used/total 406 / 504

Used/total 9

NOBACKUP2

nas_dl No errors

LustreHealth read=1705.7MB write=153.2MB

GrantSpace No errors

Free Space 384 TB (23%)

MDT/OST	Status	Mounted	Loads
S160	healthy	mounted	0.01, 0.08, 0.14
S161	healthy	mounted	1.06, 3.19, 4.23
S162	healthy	mounted	3.31, 4.14, 4.93
S163	healthy	mounted	1.44, 4.25, 5.05
S164	healthy	mounted	1.60, 2.17, 3.18
S165	healthy	mounted	1.39, 3.39, 4.60
S166	healthy	mounted	2.12, 3.25, 4.49
S167	healthy	mounted	1.09, 1.66, 2.70
S168	healthy	mounted	2.39, 2.32, 2.87

(Click to link to service group page.)

bp2



datagg (DATa AGGregator), cont.

- in-house written perl script
 - it reads the command file (pipe) that nsca creates on the webserver
 - the nagios configuration on the webserver also writes to the nsca command file
 - it aggregates the data that it reads in from the pipe and writes it out to a flat file referred to as the “HUD buffer”
 - The data read in is in the format:
[`$timestamp`] `PROCESS_SERVICE_CHECK_RESULT`;
`$hostname`; `$service_description`; `$state`; `$output`

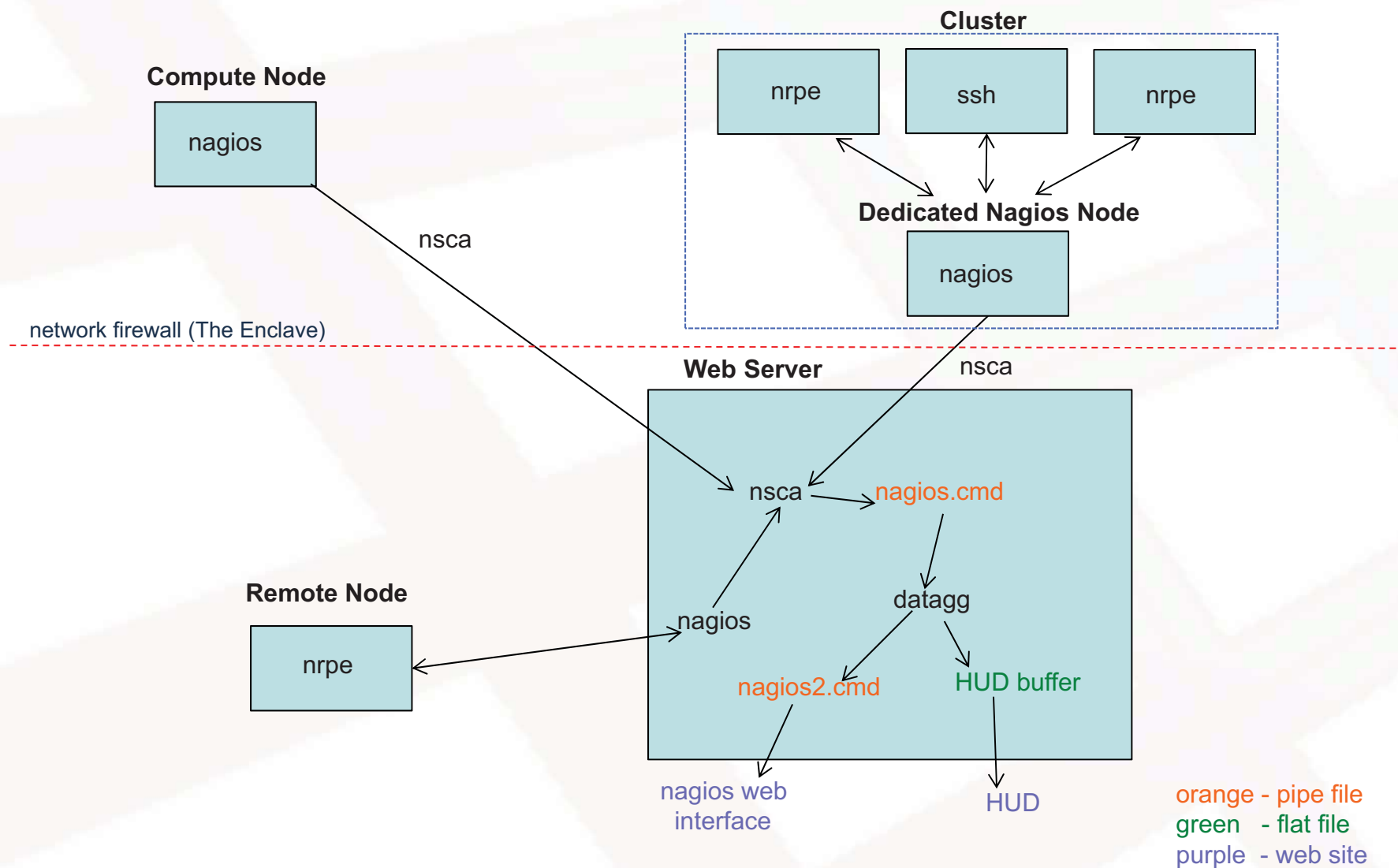


HUD Buffer

- is a Windows-style .ini file
 - sections
 - used to group together the boxes (or sub-boxes)
 - ex: [pleiades daemons]
 - keys
 - name=value
 - this is where we put the nagios state and output of plugin
 - every section also has the key Error Summary



Data flow





Displays

- Two versions
 - Internal (aka Staff HUD) on the internal network
 - Staff HUD and nagios web interface
 - clicking on Staff HUD goes to the nagios web page
 - for the service checks for the host or service group
 - the nagios web interface gives more details on the plugin output than is displayed on the HUD
 - the nagios web interface is used to suspend/restart notifications
 - External (aka miniHUD)
 - Permissions set in the Apache config file
- Both written in Perl using cgi-bin



Future Plans

- use a database to collect data
 - this will also allow us to have historical data beyond what is in the logs, which can be used in graphing
 - will eliminate the need for a flat file



Future Plans

- things that would be great to see in nagios 4
 - an API
 - make a nagios check run “on demand”
 - no more random “could not open pipe” errors
 - lower latency (we’re having problems with less than 600 checks!)
 - error summaries
 - way to send large amounts of data via nsca
 - a way to see the exact command that nagios ran on the nagios webpage

